

Lista de exercícios

Exercício 1: Clientes e contas

1. Definir uma nova classe Cliente contendo as seguintes propriedades:

- nome
- sobrenome

2. Definir uma nova classe Conta contendo as seguintes propriedades:

- número da conta
- saldo
- titular (o titular é um objeto do tipo Cliente).

3. Criar uma função chamado depósito que tenha como parâmetro de entrada quantia em dinheiro. Como estamos fazendo um depósito, a quantia de dinheiro é somada ao saldo. A função deve imprimir na tela o tipo de transação realizada e o novo saldo.

4. Criar uma função chamado saque que tenha como parâmetro de entrada quantia em dinheiro. Como estamos fazendo um saque, a quantia é subtraída do saldo. Caso o valor do saque a realizar seja maior que o saldo disponível, imprimir na tela: “Saldo insuficiente”. Caso contrário, a função deve imprimir na tela o tipo de transação realizada e o novo saldo.

Lista de exercícios

5. Definir uma nova classe chamada Exercicio1 e criar uma função main. Dentro da função main, criar dois clientes, passando nome e sobrenome, e criar uma conta para cada um, inicializando número da conta, saldo e titular. Em seguida, fazer um depósito e um saque em cada conta.

Exercício 2: Jogadores e treinamento

1. Definir uma nova classe JogadorDeFutebol contendo as seguintes propriedades:

- nome
- energia
- alegria
- gols
- experiência

2. Criar uma função chamado fazerGol em jogador, que tire 5 pontos de energia e dê 10 pontos de alegria ao jogador, além de aumentar o número de gols em 1. O método deve imprimir “GOOOOL!” na tela.

3. Criar uma função chamado correr em jogador, que tire 10 pontos de energia dele. O método deve imprimir na tela: “Cansei”.

4. Definir uma nova classe `SessaoDeTreinamento` contendo as seguintes propriedades:

- experiência (ganha quando um treinamento é realizado)

5. Criar uma função chamado `treinarA` que tenha como parâmetro de entrada um jogador de futebol. Como é um treinamento, o jogador precisa:

- Correr

- Fazer um Gol

- Correr

Depois que o jogador fizer essas três coisas, a experiência dele deve aumentar, somando ponto na experiência. A função deve imprimir a experiência inicial e a experiência final do jogador.

6. Definir um arquivo chamada `Exercicio2` e criar uma função `main`. Dentro da função `main`, criar dois jogadores de futebol, configurando as propriedades necessárias, e criar uma sessão de treinamento, também com as propriedades necessárias. Os dois jogadores devem realizar um treinamento.

Exercício 3: Atletas e prova

1. Definir uma nova classe `Atleta` contendo as seguintes propriedades:

- nome

- nível

- energia

2. Definir uma nova classe Prova contendo as seguintes propriedades:

- dificuldade

- energiaNecessaria

3. Criar, na classe Prova, a função realizarProva, que tenha como parâmetro de entrada um objeto do tipo Atleta. A função deve retornar true caso o atleta possa realizar a prova e, caso contrário, false. Um atleta pode realizar a prova se tiver um nível maior ou igual à dificuldade da prova, além de energia suficiente. Caso possa realizar, a energia do atleta deve ser reduzida.

4. Definir um arquivo chamado Exercicio3 e criar uma função main. Dentro da função main, criar dois atletas, configurando as propriedades necessárias, e criar três provas para cada um, também com os atributos necessários. Depois, verificar se os atletas podem realizar essas provas.

Exercício 4: Consumo

Classe carro: Implemente uma classe chamada Carro com as seguintes propriedades:

Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque. O consumo é especificado no construtor e o nível de combustível inicial é 0.

1. Forneça um método andar() que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.

2. Forneça um método obterGasolina(), que retorna o nível atual de combustível.
3. Forneça um método adicionarGasolina(), para abastecer o tanque. Exemplo de uso:

```
meuFusca = Carro(15);           # 15 quilômetros por litro de combustível.  
meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.  
meuFusca.andar(100);           # anda 100 quilômetros.  
meuFusca.obterGasolina()       # Imprime o combustível que resta no tanque.
```

Exercício 5: Veículos usados

Queremos desenvolver um sistema informático para uma concessionária de veículos usados. Essa concessionária decidiu que venderá apenas carros. Todos os carros têm marca, modelo, ano de fabricação, cor e quilometragem.

Lista de exercícios

Como estratégia de marketing, a concessionária quer ter um registro dos clientes que comprem os veículos. O departamento de marketing da concessionária precisa saber o nome, o sobrenome e alguma informação de contato dos clientes para poder oferecer outros carros no futuro.

Por outro lado, o departamento de contabilidade da concessionária quer ter um registro das vendas realizadas.

Cada registro deve ter as seguintes informações:

-valorDaVenda

-veiculoVendido

-cliente

1. Fazer um diagrama de classe representando o modelo que queremos implementar.

2. Implementar cada classe e definir as propriedades. As classes que devem ser implementadas são:

Concessionária, Cliente, Venda e Veículo.

3. Criar os construtores para as classes Cliente, Veículo e Venda.

- Cliente terá como parâmetros: nome, sobrenome e contato.

- Veículo terá como parâmetros marca, modelo, ano de fabricação, cor e quilometragem.

- Venda terá como parâmetros cliente, veículo e valor da venda.

Lista de exercícios

4. Definir, na classe Concessionária, a função registrarVenda(veículo: Veiculo, cliente: Cliente, valor: Double), que adiciona ao registro de vendas da concessionária uma nova venda de um veículo a um cliente por um valor determinado.

5. Definir um arquivo chamado Exercicio5 e criar a função main. Dentro da função main, criar um carro, criar um cliente e registrar a venda desse veículo a esse cliente pela Concessionária.